



ELSEVIER

Signal Processing 80 (2000) 857–866

**SIGNAL  
PROCESSING**

www.elsevier.nl/locate/sigpro

# Stable IIR digital differentiator design using iterative quadratic programming approach

Chien-Cheng Tseng

*Department of Computer and Communication Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan*

Received 16 August 1999; received in revised form 18 November 1999

---

## Abstract

In this paper, we present an iterative quadratic programming approach to design stable IIR digital differentiator. At each iteration, the cost function is transformed into a quadratic form by treating the denominator polynomial obtained from the preceding iteration as a part of the weighting function, and the pole radii are constrained to lie in the unit circle by using the implications of Rouché's theorem. After solving the standard quadratic programming problem at each iteration, the design algorithm converges to a stable and truly weighted least-squares solution. Design examples demonstrate that our method provides a better design results than the conventional quadratic programming method. © 2000 Elsevier Science B.V. All rights reserved.

## Zusammenfassung

In dieser Arbeit stellen wir ein iteratives Verfahren der quadratischen Programmierung vor, um stabile digitale IIR Differenzierer zu entwerfen. Bei jeder Iteration wird die Kostenfunktion in eine quadratische Form dadurch transformiert, daß das aus der vorhergegangenen Iteration erhaltene Nennerpolynom als Teil der Gewichtsfunktion behandelt wird, und die Polradien durch die Ausnutzung des Rouchéschen Theorems auf das Innere des Einheitskreises beschränkt werden. Nach der Lösung des gewöhnlichen Problems der quadratischen Programmierung innerhalb jeder Iteration konvergiert der Entwurfsalgorithmus zu einer stabilen und echten gewichteten Kleinstequadratlösung. Entwurfsbeispiele zeigen, daß die Methode bessere Ergebnisse liefert als die gewöhnliche Methode der quadratischen Programmierung. © 2000 Elsevier Science B.V. All rights reserved.

## Résumé

Nous présentons dans cet article une approche de programmation quadratique itérative pour la conception de différenciateurs numériques IIR stables. A chaque itération, la fonction de coût est transformée en une forme quadratique en considérant le polynôme du dénominateur obtenu à la précédente itération comme partie de la fonction de pondération, et les pôles sont contraints à se situer à l'intérieur du cercle unité à l'aide des implications du théorème de Rouché. Après résolution du problème standard de programmation quadratique à chaque itération, l'algorithme de conception converge vers une solution aux moindres carrés réellement stable et réellement pondérée. Des exemples de conception montrent que notre méthode fournit de meilleurs résultats de conception que la méthode par programmation quadratique conventionnelle. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords:* Differentiators; Quadratic programming

---

*E-mail address:* tcc@ccms.nkfu.edu.tw (C.-C. Tseng)

0165-1684/00/\$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.

PII: S0165-1684(99)00167-X

## 1. Introduction

Conventionally, digital differentiator (DD) is a very useful tool to determine and estimate the time derivatives of given signals. For example, in radars and sonars, the velocity and acceleration are computed from the position measurements using differentiators [15]. In biomedical engineering, it is often necessary to obtain the higher order derivatives of biomedical data, especially at low-frequency ranges [20]. In image processing, the derivatives at high frequencies are useful for detecting the edges [9]. So far, several methods have been developed to design FIR and IIR digital differentiators. An excellent survey of the differentiators design has been presented in tutorial paper [6].

During the past three decades, FIR digital differentiator have been designed by various methods. The earliest attempt to design an FIR differentiator was by Kaisor who used the Fourier series method for designing a wideband DD [2]. However, the Fourier series is well known for its slow convergence and truncation of the Fourier series gives rise to Gibbs phenomenon. To improve the performance, various optimization techniques are developed to obtain efficient FIR DDs such as Remez exchange algorithm [12] and eigenfilter approach [13]. Recently, the methods based on least-squares error criterion are used to design high-order FIR DD satisfying prescribed specification [17,18].

On the other hand, IIR digital differentiators have been designed by using linear and nonlinear programming methods [4,14,19]. In [14] the coefficients of IIR DD are obtained by minimizing a square error. However, this method cannot be used to design DD with constant group delay response. In [4] linear programming (LP) algorithm is employed to design DD with the prescribed magnitude and group-delay response. Unfortunately, this method requires considerable computing time and a large memory space. In [19] a quadratic programming (QP) technique is developed to design a stable IIR DD. Compared to the LP method, QP approach has a slower computational complexity and smaller variation of magnitude and group-delay errors. However, there are two disadvantages in this QP method. One is that the stability constraint is a sufficient one which is too restrictive, the

other is that an approximation is made in the derivation of the cost function. Thus, the performance of QP method in [19] can be improved by removing these two drawbacks. This is the purpose of this paper.

This paper is organized as follows. First, the design problem of IIR digital differentiator is described in Section 2. Then, the conventional QP method is reviewed briefly in Section 3. Two drawbacks of this method are also pointed out. One concerns the approximation of the cost function, the other concerns the stability constraint. In Section 4, an iterative QP method is proposed to eliminate the drawbacks of conventional QP method. Finally, several design examples are demonstrated to illustrate the effectiveness of the proposed approach.

## 2. Problem statement

The transfer function of IIR digital filter can be written as

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}}$$

$$= \frac{\mathbf{q}_2^t(z)\mathbf{b}}{1 + \mathbf{q}_1^t(z)\mathbf{a}}, \quad (1)$$

where  $\mathbf{a} = [a_1 \dots a_N]^t$ ,  $\mathbf{b} = [b_0 \ b_1 \ \dots \ b_N]^t$ ,  $\mathbf{q}_1(z) = [z^{-1} \ \dots \ z^{-N}]^t$ , and  $\mathbf{q}_2(z) = [1 \ z^{-1} \ \dots \ z^{-N}]^t$ . Now, the problem is to find the filter coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$  such that the frequency response  $H(e^{j\omega})$  fits the following  $r$ th-order differentiator response  $D(e^{j\omega})$  as close as possible:

$$D(e^{j\omega}) = M(\omega)e^{j\phi(\omega)}. \quad (2)$$

The magnitude response  $M(\omega)$  is given by

$$M(\omega) = \left(\frac{\omega}{\pi}\right)^r \quad (3)$$

for  $0 \leq \omega \leq \omega_p$ , where  $\omega_p$  is the upper passband edge frequency. The phase response  $\phi(\omega)$  is given

by

$$\phi(\omega) = \frac{r\pi}{2} - \tau_d \omega, \tag{4}$$

where  $\tau_d$  is a prescribed constant group delay whose value is set to be between  $N - 1$  and  $N/2$ . With a given weighting function  $W(\omega)$ , the weighted squares error between the desired and actual frequency responses can be defined as

$$J(\mathbf{a}, \mathbf{b}) = \int_R W(\omega) |D(e^{j\omega}) - H(e^{j\omega})|^2 d\omega, \tag{5}$$

where the integral region is given by  $R = [0, \omega_p]$ . As can be seen, the minimization of  $J(\mathbf{a}, \mathbf{b})$  leads to a nonlinear problem, so there are multiple minimum points of the cost function  $J(\mathbf{a}, \mathbf{b})$ . Thus, different minimum points may be obtained from different initial points when nonlinear programming methods are used to find the optimal solution. Moreover, to obtain a stable IIR differentiator  $H(z)$ , stability constraints must be imposed on the coefficient vector  $\mathbf{a}$ . That is, the design problem becomes

Minimize  $J(\mathbf{a}, \mathbf{b})$   
 Subject to The poles of  $H(z)$  are inside unit circle. (6)

Reduce the cost function  $J(\mathbf{a}, \mathbf{b})$  to be a quadratic form and linearize the stability constraint, this problem can be solved by a quadratic programming (QP) approach which is better than linear programming method in terms of the computational complexity and the variation of the magnitude and group-delay errors [19]. In this paper, an improved quadratic programming approach is presented to solve this design problem. Experiments will be made to show that the new QP method is better than the conventional QP method in terms of squares error defined in Eq. (5).

### 3. Conventional quadratic programming method

In this section, a quadratic programming method developed in [19] for designing recursive differentiators with constant group-delay characteristics will

be first reviewed briefly. Then, two remarks about this method are made. The cost function  $J(\mathbf{a}, \mathbf{b})$  in Eq. (5) can be expressed as

$$\begin{aligned} J(\mathbf{a}, \mathbf{b}) &= \int_R W(\omega) \left| D(e^{j\omega}) - \frac{B(e^{j\omega})}{A(e^{j\omega})} \right|^2 d\omega \\ &= \int_R \frac{W(\omega)}{|A(e^{j\omega})|^2} |D(e^{j\omega})A(e^{j\omega}) - B(e^{j\omega})|^2 d\omega. \end{aligned} \tag{7}$$

In [19] neglect the term  $|A(e^{j\omega})|^2$  underneath  $W(\omega)$  in Eq. (7), leading the following modified cost function:

$$\begin{aligned} \hat{J}(\mathbf{a}, \mathbf{b}) &= \int_R W(\omega) |D(e^{j\omega})A(e^{j\omega}) - B(e^{j\omega})|^2 d\omega \\ &= \int_R W(\omega) |D(e^{j\omega})(1 + \mathbf{q}_1^t(e^{j\omega})\mathbf{a}) \\ &\quad - \mathbf{q}_2^t(e^{j\omega})\mathbf{b}|^2 d\omega \\ &= \int_R W(\omega) |D(e^{j\omega}) + \Phi^t(e^{j\omega})\mathbf{x}|^2 d\omega, \end{aligned} \tag{8}$$

where

$$\Phi(e^{j\omega}) = \begin{bmatrix} D(e^{j\omega})\mathbf{q}_1^t(e^{j\omega}) \\ -\mathbf{q}_2^t(e^{j\omega}) \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}.$$

After some manipulation, Eq. (8) can be written as

$$\hat{J}(\mathbf{x}) = \mathbf{x}^t \mathbf{Q} \mathbf{x} + 2\mathbf{p}^t \mathbf{x} + c, \tag{9}$$

where

$$\begin{aligned} \mathbf{Q} &= \int_R W(\omega) \text{Re}(\Phi(e^{j\omega})\Phi^H(e^{j\omega})) d\omega, \\ \mathbf{p} &= \int_R W(\omega) \text{Re}(D^*(e^{j\omega})\Phi(e^{j\omega})) d\omega, \\ c &= \int_R W(\omega) |D(e^{j\omega})|^2 d\omega. \end{aligned}$$

In [19] in order to make IIR filter  $H(z)$  be stable, a stability constraint imposed on the denominator polynomial  $A(z)$  is given by

$$\text{Re}(A(z)) \geq \delta \quad \text{for } |z| = 1, \tag{10}$$

where  $\text{Re}(A(z))$  denotes the real part of  $A(z)$  and  $\delta$  is a small positive number. From Eqs. (1) and (10), we

obtain

$$- \mathbf{a}^t \operatorname{Re}(\mathbf{q}_1(e^{j\theta})) \leq 1 - \delta \quad \text{for } 0 \leq \theta \leq \pi. \quad (11)$$

Let  $\Theta = \{\theta_i \mid i = 1, \dots, L\}$  be the set of grid points on  $[0, \pi]$ . Using matrix notation, condition Eq. (11) on set  $\Theta$  becomes

$$\mathbf{G}\mathbf{x} \leq (1 - \delta)\mathbf{e}, \quad (12)$$

where

$$\mathbf{G} = \begin{bmatrix} -\operatorname{Re}(\mathbf{q}_1^t(e^{j\theta_1})) & \mathbf{0} \\ -\operatorname{Re}(\mathbf{q}_1^t(e^{j\theta_2})) & \mathbf{0} \\ \vdots & \vdots \\ -\operatorname{Re}(\mathbf{q}_1^t(e^{j\theta_L})) & \mathbf{0} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Thus, the design problem becomes a standard quadratic programming (QP) problem below:

$$\text{Minimize } \mathbf{x}^t \mathbf{Q}\mathbf{x} + 2\mathbf{p}^t \mathbf{x} + c$$

$$\text{Subject to } \mathbf{G}\mathbf{x} \leq (1 - \delta)\mathbf{e}. \quad (13)$$

With a positive definite  $\mathbf{Q}$ , the solution of QP problem can be computed efficiently. For details one can refer to [11]. Now, two remarks about this method are made as follows. One concerns the cost function  $\hat{J}(\mathbf{a}, \mathbf{b})$ , the other concerns the stability constraint.

**Remark 1.** From cost function  $J(\mathbf{a}, \mathbf{b})$  to  $\hat{J}(\mathbf{a}, \mathbf{b})$ , the term  $|A(e^{j\omega})|^2$  underneath  $W(\omega)$  is neglected. Thus, the rational function  $B(z)/A(z)$  that minimize  $\hat{J}(\mathbf{a}, \mathbf{b})$  does not necessarily minimize  $J(\mathbf{a}, \mathbf{b})$ ; hence, the solution obtained by solving Eq. (13) is, in general, not optimal in the least-squares sense. In next section, an iterative QP approach will be proposed to remove this drawback.

**Remark 2.** In Eq. (11), a constraint that is linear w.r.t. the denominator polynomial coefficient  $\mathbf{a}$  is used to guarantee stability. The severe drawback of these linear constraints is the fact that they are sufficient but not necessary for stability. In numerous experiments, we were always able to find better stable solutions that do not satisfy these constraints. Recently, an interesting and relevant development about this problem is the use of Rouché's theorem for the design of stable IIR filters. In the

next section, we will use Rouché's theorem to solve the stability problem.

#### 4. Iterative quadratic programming method

In this section, an iterative QP method is presented to design IIR differentiator such that the performance of conventional QP method can be improved.

##### 4.1. Cost function

Instead of neglecting the term  $|A(e^{j\omega})|^2$  underneath  $W(\omega)$  in Eq. (7), the optimization problem can be solved by the following iterative scheme:

$$J_k(\mathbf{a}_k, \mathbf{b}_k) = \int_R \frac{W(\omega)}{|A_{k-1}(e^{j\omega})|^2} |D(e^{j\omega})A_k(e^{j\omega}) - B_k(e^{j\omega})|^2 d\omega, \quad (14)$$

where  $A_k(e^{j\omega}) = 1 + \mathbf{q}_1^t(e^{j\omega})\mathbf{a}_k$ ,  $B_k(e^{j\omega}) = \mathbf{q}_2^t(e^{j\omega})\mathbf{b}_k$ ,  $\mathbf{a}_k$  and  $\mathbf{b}_k$  are the parameter vectors to be determined in the  $k$ th iteration. The iterative scheme in Eq. (14) has also been used in Eq. (5) of Ref. [10]. The initial vector  $\mathbf{a}_0$  can be chosen quite arbitrarily, except that  $A_0(z)$  must be stable. For example, one may choose  $\mathbf{a}_0 = [0, 0, \dots, 0]^t$ . At the  $k - 1$  iteration,  $A_{k-1}(e^{j\omega})$  is known and stable, hence, Eq. (14) can be rewritten as

$$J_k(\mathbf{x}_k) = \int_R W_{k-1}(\omega) |D(e^{j\omega}) + \Phi^t(e^{j\omega})\mathbf{x}_k|^2 d\omega, \quad (15)$$

where

$$\Phi(e^{j\omega}) = \begin{bmatrix} D(e^{j\omega})\mathbf{q}_1(e^{j\omega}) \\ -\mathbf{q}_2(e^{j\omega}) \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} \mathbf{a}_k \\ \mathbf{b}_k \end{bmatrix},$$

$$W_{k-1}(\omega) = \frac{W(\omega)}{|A_{k-1}(e^{j\omega})|^2}. \quad (16)$$

After some manipulation, Eq. (15) can be written as

$$J_k(\mathbf{x}_k) = \mathbf{x}_k^t \mathbf{Q}_k \mathbf{x}_k + 2\mathbf{p}_{k-1}^t \mathbf{x}_k + c_{k-1}, \quad (17)$$

where

$$\begin{aligned} \mathbf{Q}_{k-1} &= \int_R W_{k-1}(\omega) \text{Re}(\Phi(e^{j\omega})\Phi^H(e^{j\omega})) d\omega, \\ \mathbf{P}_{k-1} &= \int_R W_{k-1}(\omega) \text{Re}(D^*(e^{j\omega})\Phi(e^{j\omega})) d\omega, \\ c_{k-1} &= \int_R W_{k-1}(\omega) |D(e^{j\omega})|^2 d\omega. \end{aligned} \tag{18}$$

4.2. Stability issue

The rational function generated from the  $k$ th iteration must be stable. Thus, some stability constraint must be imposed on the polynomial  $A_k(z)$ . Although to transform the stability constraint into a set of linear constraints is a difficult problem, Lang has proposed an interesting method to solve this problem recently [7]. This method is mainly based on Rouché's theorem:

**Rouché's theorem.** *If  $f(z)$  and  $g(z)$  are analytic inside and on a closed contour  $C$ , and  $|g(z)| < |f(z)|$  on  $C$ , then  $f(z)$  and  $f(z) + g(z)$  have the same number of zeros inside  $C$ .*

A proof of Rouché's theorem can be found in [5]. Now, let the contour  $C$  be the unit circle and the denominator polynomial  $A_{k-1}(z)$  at iteration step  $k-1$  has all its zeros inside unit circle, then, according to Rouché's theorem, the denominator polynomial in the next iteration  $k$  is given by

$$A_k(z) = A_{k-1}(z) + \alpha \Delta_k(z), \quad 0 < \alpha < 1 \tag{19}$$

will also have all its zeros inside unit circle if the following condition is satisfied:

$$|\Delta_k(z)| \leq |A_{k-1}(z)| - \delta, \quad |z| = 1, \tag{20}$$

where  $\delta$  is a small positive number. Since two polynomials  $A_k(z)$  and  $\Delta_k(z)$  are given by

$$\begin{aligned} \Delta_k(z) &= \mathbf{d}_k^t \mathbf{q}_1(z), \\ A_k(z) &= 1 + \mathbf{a}_k^t \mathbf{q}_1(z), \end{aligned} \tag{21}$$

where  $\mathbf{q}_1(z)$  has been defined in Eq. (1) and

$$\begin{aligned} \mathbf{a}_k &= [a_1, a_2, \dots, a_N]^t, \\ \mathbf{d}_k &= [d_1, d_2, \dots, d_N]^t, \end{aligned}$$

so Eq. (19) can be rewritten as

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha \mathbf{d}_k. \tag{22}$$

Substitute Eq. (22) into  $\mathbf{x}_k$  in Eq. (16), we obtain

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} \mathbf{a}_{k-1} + \alpha \mathbf{d}_k \\ \mathbf{b}_k \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{k-1} \\ \mathbf{0} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{d}_k \\ \frac{1}{\alpha} \mathbf{b}_k \end{bmatrix} \\ &= \mathbf{u}_{k-1} + \alpha \mathbf{y}_k, \end{aligned} \tag{23}$$

where vectors  $\mathbf{u}_{k-1}$  and  $\mathbf{y}_k$  are defined by

$$\mathbf{u}_{k-1} = \begin{bmatrix} \mathbf{a}_{k-1} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} \mathbf{d}_k \\ \frac{1}{\alpha} \mathbf{b}_k \end{bmatrix}. \tag{24}$$

Now, let us study the constraints in Eq. (20). Let  $\Theta = \{\theta_i \mid i = 1, \dots, L\}$  be the set of dense grid points on  $[0, \pi]$ , then this constraint, on the set  $\Theta$ , reduces to

$$|\Delta_k(e^{j\theta_i})| \leq |A_{k-1}(e^{j\theta_i})| - \delta, \quad i = 1, \dots, L. \tag{25}$$

Let  $g_i = |A_{k-1}(e^{j\theta_i})| - \delta$ , the vectors

$$\mathbf{r}_i = \text{Real}(\mathbf{q}_1(e^{j\theta_i})),$$

$$\mathbf{s}_i = \text{Imag}(\mathbf{q}_1(e^{j\theta_i}))$$

and two scalars

$$v_i = \mathbf{d}_k^t \mathbf{r}_i, \quad z_i = \mathbf{d}_k^t \mathbf{s}_i, \tag{26}$$

then Eq. (25) becomes

$$\sqrt{v_i^2 + z_i^2} \leq g_i, \quad i = 1, \dots, L. \tag{27}$$

Obviously, it is a quadratic constraint of the coefficients  $\mathbf{d}_k$ . Now, this quadratic constraint can be converted to the linear constraints by approximating a circle with an octagon shown in Fig. 1. This approximation technique has also been used in the design of FIR filters in the complex domain [3]. Thus, the quadratic constraint in Eq. (27) can be approximated by the following linear constraints:

$$\begin{aligned} \left| \cos\left(\frac{n\pi}{4}\right)v_i + \sin\left(\frac{n\pi}{4}\right)z_i \right| &\leq g_i \cos\left(\frac{\pi}{8}\right), \\ n &= 0, 1, 2, 3, \quad i = 1, \dots, L. \end{aligned} \tag{28}$$

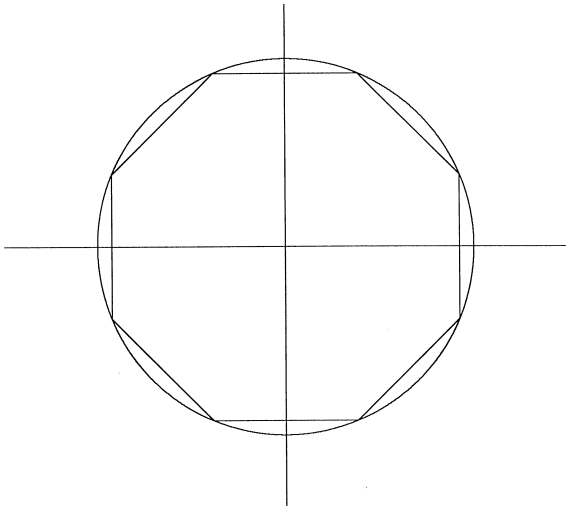


Fig. 1. Approximation of a circle with an octagon.

It is clear that if  $(v_i, z_i)$  satisfies the linear constraints in Eq. (28) simultaneously, the  $(v_i, z_i)$  must satisfy the quadratic constraint in Eq. (27). Substituting Eq. (26) into Eq. (28), we obtain

$$\mathbf{T}_i \mathbf{d}_k \leq \mathbf{h}_i, \quad i = 1, \dots, L, \tag{29}$$

where matrix  $\mathbf{T}_i$  and vector  $\mathbf{h}_i$  are given by

$$\mathbf{T}_i = \begin{bmatrix} r_i^t \\ -r_i^t \\ \left(\frac{r_i+s_i}{\sqrt{2}}\right)^t \\ -\left(\frac{r_i+s_i}{\sqrt{2}}\right)^t \\ s_i^t \\ -s_i^t \\ \left(\frac{-r_i+s_i}{\sqrt{2}}\right)^t \\ -\left(\frac{-r_i+s_i}{\sqrt{2}}\right)^t \end{bmatrix}, \quad \mathbf{h}_i = g_i \cos\left(\frac{\pi}{8}\right) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{30}$$

$i = 1, \dots, L.$

Using Eq. (24) and matrix notation, the stability condition in Eq. (29) becomes of the following linear form:

$$\mathbf{T} \mathbf{y}_k \leq \mathbf{h}, \tag{31}$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} \\ \mathbf{T}_2 & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{T}_L & \mathbf{0} \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_L \end{bmatrix}. \tag{32}$$

### 4.3. Design algorithm

After investigating the cost function and the stability issue, let us combine these two results to obtain the final design algorithm. Substituting Eq. (23) into Eq. (17), the cost function becomes

$$\begin{aligned} J_k(\mathbf{x}_k) &= \mathbf{x}_k^t \mathbf{Q}_{k-1} \mathbf{x}_k + 2\mathbf{p}_{k-1}^t \mathbf{x}_k + c_{k-1} \\ &= (\mathbf{u}_{k-1} + \alpha \mathbf{y}_k)^t \mathbf{Q}_{k-1} (\mathbf{u}_{k-1} + \alpha \mathbf{y}_k) \\ &\quad + 2\mathbf{p}_{k-1}^t (\mathbf{u}_{k-1} + \alpha \mathbf{y}_k) + c_{k-1} \\ &= \mathbf{y}_k^t \bar{\mathbf{Q}}_{k-1} \mathbf{y}_k + 2\bar{\mathbf{p}}_{k-1}^t \mathbf{y}_k + \bar{c}_{k-1}, \end{aligned} \tag{33}$$

where

$$\begin{aligned} \bar{\mathbf{Q}}_{k-1} &= \alpha^2 \mathbf{Q}_{k-1}, \\ \bar{\mathbf{p}}_{k-1} &= \alpha \mathbf{Q}_{k-1} \mathbf{u}_{k-1} + \alpha \mathbf{p}_{k-1}, \\ \bar{c}_{k-1} &= \mathbf{u}_{k-1}^t \mathbf{Q}_{k-1} \mathbf{u}_{k-1} + 2\mathbf{p}_{k-1}^t \mathbf{u}_{k-1} + c_{k-1}. \end{aligned} \tag{34}$$

Incorporate the stability constraint in Eq. (31), the design problem becomes the standard quadratic programming form as follows:

$$\begin{aligned} &\text{Minimize} \quad \mathbf{y}_k^t \bar{\mathbf{Q}}_{k-1} \mathbf{y}_k + 2\bar{\mathbf{p}}_{k-1}^t \mathbf{y}_k + \bar{c}_{k-1} \\ &\text{Subject to} \quad \mathbf{T} \mathbf{y}_k \leq \mathbf{h}. \end{aligned} \tag{35}$$

Based on the above description, we propose an iterative quadratic programming algorithm for obtaining the coefficients  $\mathbf{a}$  and  $\mathbf{b}$  as follows:

*Step 1:* Given the specification  $D(e^{j\omega})$  of digital differentiator, weighting function  $W(\omega)$ , number of grid points  $L$ , parameters  $\alpha$  and  $\delta$ .

*Step 2:* Set the initial denominator polynomial  $A_0(z) = 1$ , i.e., initial coefficient vector  $\mathbf{a}_0 = [0, 0, \dots, 0]^t$ . Moreover, set vector  $\mathbf{y}_0 = [0, 0, \dots, 0]^t$  and  $k = 1$ .

*Step 3:* Use Eq. (18) to compute  $\mathbf{Q}_{k-1}$ ,  $\mathbf{p}_{k-1}$ , and  $c_{k-1}$ .

*Step 4:* Use Eq. (34) to compute  $\bar{\mathbf{Q}}_{k-1}$ ,  $\bar{\mathbf{p}}_{k-1}$ , and  $\bar{c}_{k-1}$ .

Step 5: Use Eqs. (30) and (32) to compute the stability constraint parameters  $\mathbf{T}$  and  $\mathbf{h}$ .

Step 6: Solve the quadratic programming problem in Eq. (35) to obtain the coefficient  $\mathbf{y}_k$ .

Step 7: Obtain  $\mathbf{d}_k$  and  $\mathbf{b}_k$  from  $\mathbf{y}_k$  by using the relation in Eq. (24).

Step 8: Compute  $\mathbf{a}_k$  by using Eq. (22), i.e.,

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \alpha \mathbf{d}_k.$$

Step 9: Terminate the iterative procedure if

$$\frac{\|\mathbf{y}_k - \mathbf{y}_{k-1}\|}{\|\mathbf{y}_k\|} \leq \varepsilon,$$

where  $\varepsilon$  is a preset positive number. Otherwise, set  $k = k + 1$  and go to step 3.

Now, let us present a sufficient condition for the convergence of the proposed algorithm. Define the ratio

$$\eta_k = \frac{\|\mathbf{y}_{k+1} - \mathbf{y}_k\|}{\|\mathbf{y}_k - \mathbf{y}_{k-1}\|}. \tag{36}$$

It can be shown that if  $\eta_k$  has a less-than-unity upper bound, i.e.,

$$\eta_k \leq \gamma < 1 \tag{37}$$

for  $k \geq K$  where  $K$  is a positive integer, then sequence  $\{\mathbf{y}_k\}$  converges. As a matter of fact, this condition implies that

$$\|\mathbf{y}_{k+1} - \mathbf{y}_k\| \leq \gamma \|\mathbf{y}_k - \mathbf{y}_{k-1}\|. \tag{38}$$

So for sufficiently large  $m$  and  $n$  with  $m > n \geq K$ , we have

$$\|\mathbf{y}_m - \mathbf{y}_n\| \leq \frac{\gamma^{n-K+1} - \gamma^{m-K+1}}{1 - \gamma} \|\mathbf{y}_K - \mathbf{y}_{K-1}\| \tag{39}$$

which approaches zero when  $m, n \rightarrow \infty$ , and hence  $\{\mathbf{y}_k\}$  is a Cauchy sequence in a finite-dimensional Euclidean space. If  $\Psi$  is the operator that maps  $\mathbf{y}_k$  to  $\mathbf{y}_{k+1}$ , i.e.,  $\Psi(\mathbf{y}_k) = \mathbf{y}_{k+1}$ , then the above sufficient condition is equivalent to

$$\|\Psi(\mathbf{y}_k) - \Psi(\mathbf{y}_{k-1})\| \leq \gamma \|\mathbf{y}_k - \mathbf{y}_{k-1}\|. \tag{40}$$

In other words,  $\{\mathbf{y}_k\}$  is convergent if  $\Psi$  is a contraction mapping. Although a rigorous proof is not available to date, the proposed algorithm was al-

ways successful in producing  $\{\mathbf{y}_k\}$  with ratio  $\eta_k \leq \gamma < 1$  in our extensive simulation study.

The proposed design algorithm for the stable IIR differentiators depends on the existence of efficient algorithms to solve the quadratic programming (QP) problems which have been known for a long time. In the context of FIR filter design, several algorithms for solving the constrained least squares problem have been presented [1,8]. On the other hand, using QP to design IIR filter is also investigated recently [10,16]. In this paper, the subroutine QP.m in the optimization toolbox of MATLAB software is used to solve QP problem.

### 5. Design examples and comparisons

In this section, two examples are presented to illustrate the proposed design methodology. The weighting function chosen in the following examples is  $W(\omega) = 1$ .

**Example 1** (First-order differentiator). In this example, a first-order differentiator with the following specification is designed:

$$D(e^{j\omega}) = \left(\frac{\omega}{\pi}\right) (\sin(\tau_d \omega) + j \cos(\tau_d \omega)), \quad 0 \leq \omega \leq \omega_p, \tag{41}$$

where  $\omega_p = 0.9\pi$  and  $\tau_d = 13$ . The filter order  $N$  is set to be 15. When the iterative QP algorithm with  $\varepsilon = 10^{-4}$ , number of grid points  $L = 200$ ,  $\alpha = 0.99$  and  $\delta = 0.001$  is used to design this filter, the algorithm converges after 7 iterations. The magnitude response of this differentiator is shown in Fig. 2(a), while the group delay response is depicted in Fig. 2(b). It is clear that the specification is well satisfied in the passband  $[0, \omega_p]$ . The maximum modulus of the filter poles is 0.9981; hence the filter is stable. Moreover, the design results of conventional QP method in [19] are also shown in Fig. 3(a) and (b). In order to compare the performance of our method with conventional QP approach, a true squares error Err is defined by

$$\text{Err} = \int_0^{\omega_p} |E(\omega)|^2 d\omega, \tag{42}$$

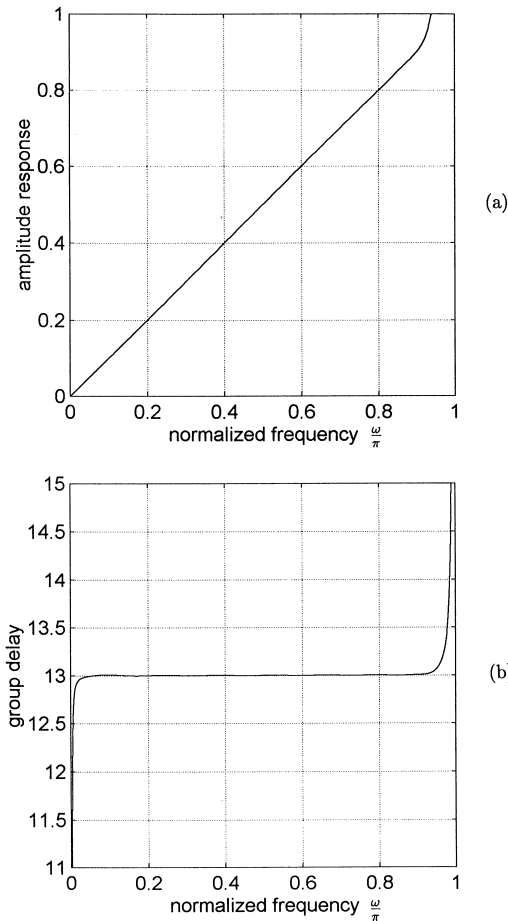


Fig. 2. Frequency response of the first-order differentiator with  $\omega_p = 0.9\pi$ ,  $N = 15$  and  $\tau_d = 13$ : (a) magnitude response designed by iterative QP method, (b) group-delay response designed by iterative QP method.

where  $|E(\omega)| = |D(e^{j\omega}) - H(e^{j\omega})|$ . The smaller the error  $\text{Err}$ , the better the design result is. In this example, the squares error  $\text{Err}$  of the proposed iterative QP method is equal to  $2.4293 \times 10^{-8}$ , and  $\text{Err}$  of the conventional QP method is  $9.1157 \times 10^{-8}$ . Thus, the proposed iterative QP method have a smaller design error than the conventional QP method. The main reason is that two drawbacks in conventional QP method have been removed in the proposed QP method.

**Example 2** (Second-order differentiator). In this example, a second-order differentiator with the

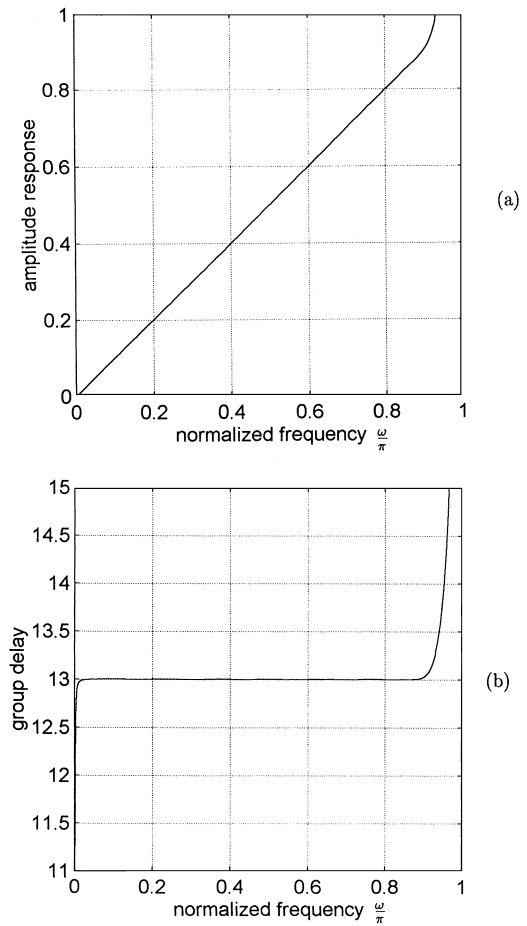


Fig. 3. Frequency response of the first-order differentiator with  $\omega_p = 0.9\pi$ ,  $N = 15$  and  $\tau_d = 13$ : (a) magnitude response designed by conventional QP method, (b) group-delay response designed by conventional QP method.

following specification is designed:

$$D(e^{j\omega}) = -\left(\frac{\omega}{\pi}\right)^2 (\cos(\tau_d \omega) - j \sin(\tau_d \omega)),$$

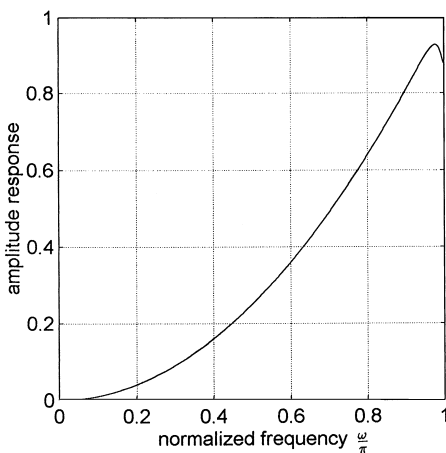
$$0 \leq \omega \leq \omega_p, \quad (43)$$

where  $\tau_d = 14$  and  $\omega_p = 0.95\pi$ . The filter order  $N$  is set to be 17. When the iterative QP algorithm with  $\varepsilon = 10^{-4}$ ,  $L = 200$ ,  $\alpha = 0.99$  and  $\delta = 0.001$  is used to design this filter, the algorithm converges after 7 iterations. The magnitude response of this differentiator is shown in Fig. 4(a), while the group delay response is depicted in Fig. 4(b). It is clear that the

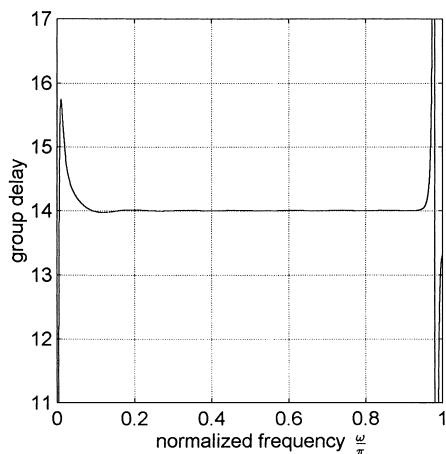
Table 1

The number of iteration of convergence, Err, and the maximum pole modulus for various parameters  $L$ ,  $\alpha$ , and  $\delta$ 

$L$	$\alpha$	$\delta$	Number of iteration of convergence	Err	Pole modulus
100	0.9	0.01	9	$1.4793 \times 10^{-7}$	0.9628
		0.001	9	$4.1695 \times 10^{-8}$	0.9965
	0.99	0.01	8	$7.9788 \times 10^{-8}$	0.9463
		0.001	8	$1.9276 \times 10^{-8}$	0.9900
200	0.9	0.01	8	$1.4580 \times 10^{-7}$	0.9642
		0.001	8	$3.8233 \times 10^{-8}$	0.9989
	0.99	0.01	7	$7.8945 \times 10^{-8}$	0.9482
		0.001	7	$1.8890 \times 10^{-8}$	0.9896



(a)



(b)

Fig. 4. Frequency response of the second-order differentiator with  $\omega_p = 0.95\pi$ ,  $N = 17$  and  $\tau_d = 14$ : (a) magnitude response, (b) group-delay response.

specification is well satisfied in the passband  $[0, \omega_p]$ . The maximum modulus of the filter poles is 0.9896; hence the filter is stable. The resultant squares error Err in this design is equal to  $1.8890 \times 10^{-8}$ . Moreover, the squares error Err designed by the conventional QP method in [19] is  $3.7392 \times 10^{-7}$ . Thus, the proposed iterative QP method have the smaller design error than the conventional QP method.

In order to investigate the effect of the parameters  $L$ ,  $\alpha$  and  $\delta$  on the designed results, Table 1 lists the number of iteration of convergence, Err and the maximum pole modulus for various parameters. It is clear that the errors Err are all smaller than the Err ( $3.7392 \times 10^{-7}$ ) of the conventional QP method. There are three observations as follows: (1) When  $\alpha$  and  $\delta$  are specified, the larger  $L$  is preferred. This is because  $L$  is introduced to convert the semi-infinite programming problem into a finite programming problem. The larger the  $L$ , the better the approximation is. (2) When  $L$  and  $\alpha$  are chosen, the larger  $\delta$ , the smaller the maximum pole modulus is. Thus, the parameter  $\delta$  can be used to control the maximum pole modulus. (3) When  $L$  and  $\delta$  are specified, the larger the  $\alpha$ , the faster the convergence speed is. This is because  $\alpha$  is a step size to update  $A_{k-1}(z)$  into  $A_k(z)$  (see Eq. (19)).

## 6. Conclusion

In this paper, an iterative quadratic programming approach has been presented to design stable

IIR digital differentiator. At each iteration, the cost function is transformed into a quadratic form by treating the denominator polynomial obtained from the preceding iteration as a part of the weighting function, and the pole radii are constrained to lie in the unit circle by using the implications of Rouché's theorem. After solving the standard quadratic programming problem at each iteration, the design algorithm converges to a stable and truly weighted least-squares solution. Design examples demonstrate that our method provides better design results than the conventional quadratic programming method.

## References

- [1] J.W. Adams, J.L. Sullivan, Peak-constrained least-squares optimization, *IEEE Trans. Signal Process.* 46 (February 1998) 306–321.
- [2] A. Antoniou, *Digital Filters: Analysis, Design and Applications*, 2nd Edition, McGraw-Hill, New York, 1993.
- [3] X. Chen, T.W. Parks, Design of FIR filters in the complex domain, *IEEE Trans. Acoust. Speech Signal Process.* 35 (February 1987) 144–153.
- [4] A.T. Chottera, G.A. Jullien, A linear programming approach to recursive filter design with linear phase, *IEEE Trans. Circuits Syst.* 29 (March 1982) 139–149.
- [5] R.V. Churchill, J.W. Brown, *Complex Variables and Applications*, McGraw-Hill, New York, 1984.
- [6] S.C. Dutta Roy, B. Kumar, Digital differentiators, in: N.K. Bose, C.R. Rao (Eds.), *Handbook of Statistics*, Vol. 10, Elsevier, Amsterdam, 1993, pp. 159–205.
- [7] M.C. Lang, Weighted least-squares IIR filter design with arbitrary magnitude and phase responses and specified stability margin, in: *Proceedings of the 1998 IEEE Symposium on Advances in Digital Filtering and Signal Processing*, Victoria, Canada, June 1998, pp. 82–86.
- [8] M. Lang, I.W. Selesnick, C.S. Burrus, Constrained least squares design of 2-D FIR filters, *IEEE Trans. Signal Process.* 44 (May 1996) 1234–1241.
- [9] J.S. Lim, *Two Dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [10] W.S. Lu, S.C. Pei, C.C. Tseng, A weighted least-squares method for the design of stable 1-D and 2-D IIR digital filters, *IEEE Trans. Signal Process.* 46 (January 1998) 1–10.
- [11] D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd Edition, Addison-Wesley, Reading, MA, 1984.
- [12] T.W. Parks, J.H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, *IEEE Trans. Circuit Theory* 19 (March 1972) 189–194.
- [13] S.C. Pei, J.J. Shyu, Eigenfilter design of higher-order digital differentiators, *IEEE Trans. Acoust. Speech Signal Process.* 37 (1989) 505–511.
- [14] L.R. Rabiner, K. Steiglitz, The design of wideband recursive and nonrecursive digital differentiators, *IEEE Trans. Audio Electroacoust.* 18 (June 1970) 204–209.
- [15] M.I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill, New York, 1980.
- [16] J.L. Sullivan, J.W. Adams, PCLS IIR digital filters with simultaneous frequency response magnitude and group delay specifications, *IEEE Trans. Signal Process.* 46 (November 1998) 2853–2861.
- [17] S. Sunder, W.S. Lu, A. Antoniou, Y. Su, Design of digital differentiators satisfying prescribed specifications using optimization techniques, *Proc. Inst. Elect. Eng., Part G* 138 (June 1991) 315–320.
- [18] S. Sunder, V. Ramachandra, Design of equiripple nonrecursive digital differentiators and Hilbert transformers using a weighted least-squares technique, *IEEE Trans. Signal Process.* 42 (9) (September 1994) 2504–2509.
- [19] S. Sunder, V. Ramachandran, Design of recursive differentiators with constant group delay characteristics, *Signal Processing* 39 (September 1994) 79–88.
- [20] S. Usui, I. Amidror, Digital lowpass differentiation for biological signal processing, *IEEE Trans. Biomed. Eng.* 29 (1982) 686–693.